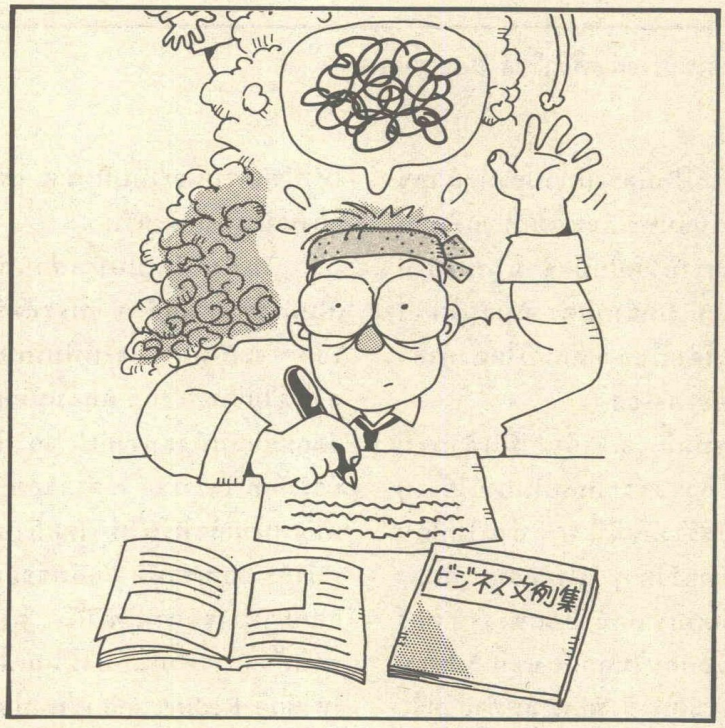
ประดิษฐ์ ภิญโญภาสกุล

SOFT MAIL

เทคนิคการเขียนโปรแกรมแสดงผลภาษาไทยในเท็กซ์โหมด

แนะนำวิธีการเขียนโปรแกรมสั้น ๆ เพื่อแสดงผลภาษาไทยในเท็กซ์โหมด สำหรับจอวีจีเอและอีจีเอ



คงมีบางครั้งที่ท่านผู้อ่านต้องการแสดงข้อความภาษาไทยในโปรแกรมที่เขียนขึ้น แต่ก็ต้องประสบกับปัญหาว่า ถ้าไม่ต้องการแสดงผลในกราฟิกโหมดก็จะต้องให้ผู้ใช้โหลดไดรเวอร์ภาษาไทยก่อนรันโปรแกรม หรือหากจะเปลี่ยนไปแสดงผลในกราฟิกโหมด การเขียนโปรแกรมก็จะยุ่งยากขึ้น อีกทั้งโปรแกรมยังมีขนาดใหญ่ขึ้นและรันได้ช้าลงอีกด้วย ทั้ง ๆ ที่ในความเป็นจริงแล้วเราอาจจะต้องการแสดงข้อความภาษาไทยเพียงไม่กี่ประโยคเท่านั้น

แต่ด้วยการอาศัยความสามารถของจอภาพแบบอีจีเอหรือวีจีเอ ที่เราสามารถเปลี่ยนบิตแม็พของฟอนต์สำหรับการแสดงผลทั้งในเท็กซ์โหมดและกราฟิกโหมดได้ตามต้องการ จึงทำให้ผู้เขียนเกิดความคิดที่จะพัฒนารูทีนสำหรับการแสดงผลภาษาไทยในเท็กซ์โหมดขึ้น โดยรูทีนที่เขียนขึ้นเป็นรูทีนสั้น ๆ ที่ใช้ได้กับจอวีจีเอเท่านั้น อย่างไรก็ดี ผู้ที่ใช้จออีจีเอก็สามารถนำรูทีนดังกล่าวไปดัดแปลงใช้งานได้เช่นกัน

ต่อไปเราจะมาศึกษาเครื่องมือที่เราจะนำมาใช้ในรูทีนของเรากันครับ

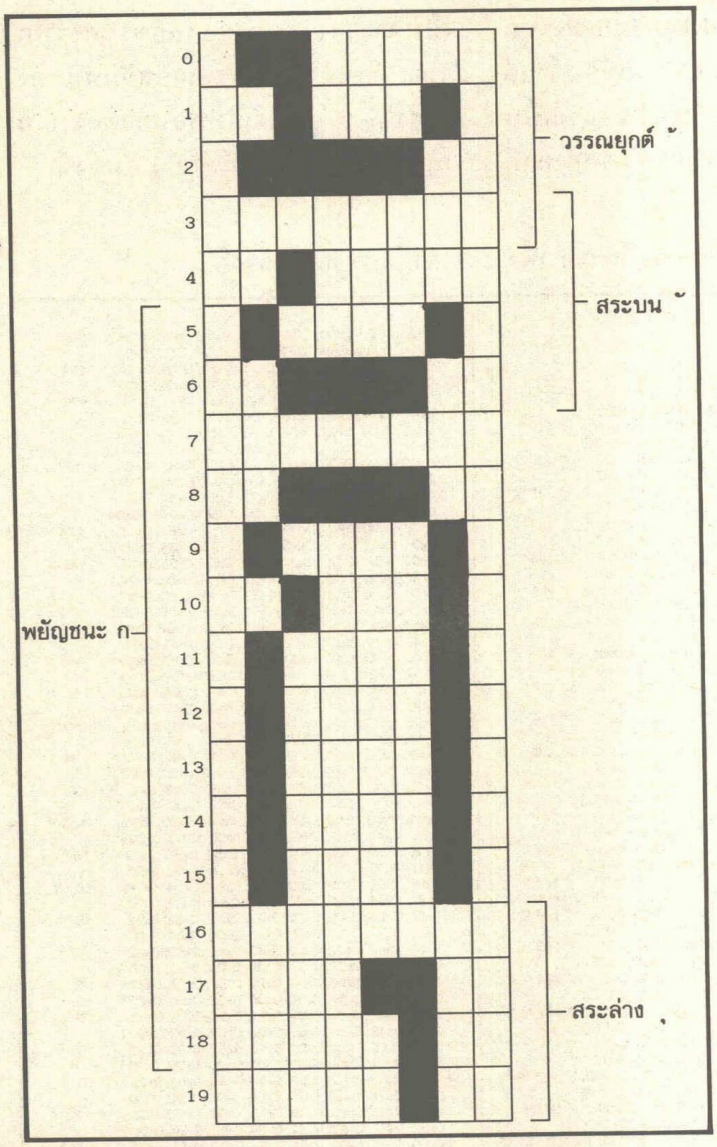
ลักษณะของบิตแม็พ

เริ่มแรกเรามาศึกษาลักษณะของบิตแม็พในตารางบิตแม็พที่เราจะใช้กันก่อน ตารางบิตแม็พสำหรับเท็กซ์โหมดของจออีจีเอและวีจีเอมีอยู่ 4 ตาราง (หมายเลข 0-3) ตารางปกติที่จะถูกใช้คือตารางหมายเลข 0 ขนาดของบิตแม็พสำหรับจออีจีเอและจอวีจีเอจะมีความแตกต่างกัน โดยจออีจีเอจะมีขนาด 14 ไบต์ และจอวีจีเอจะมีขนาด 16 ไบต์

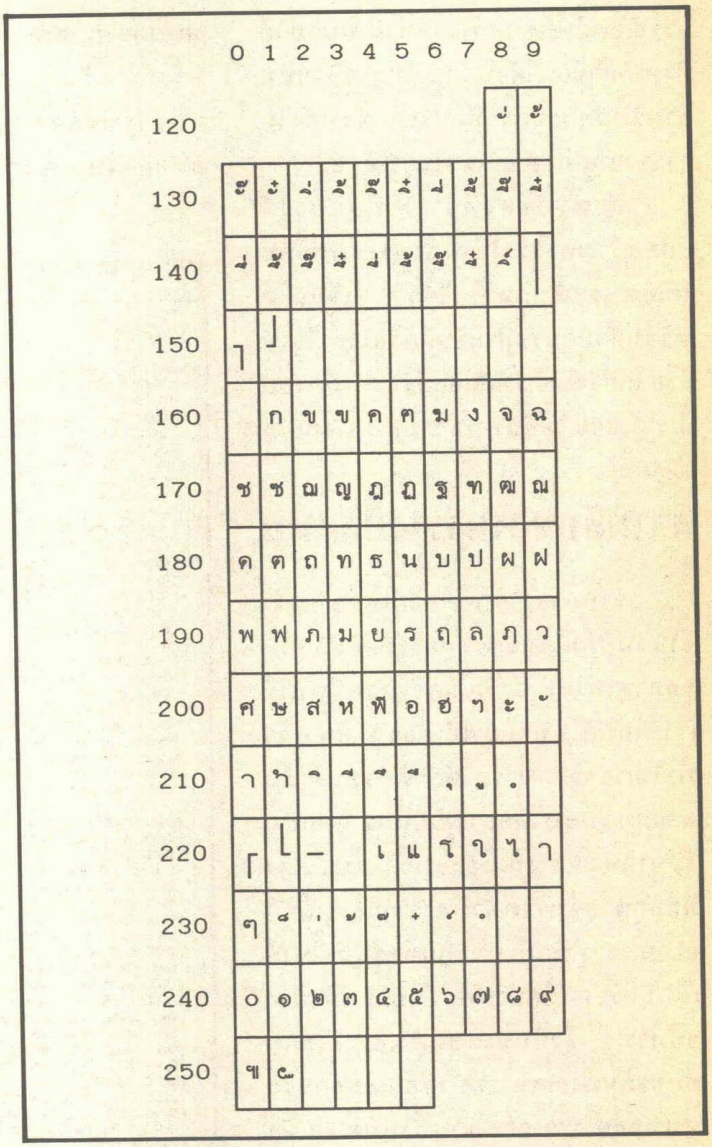
เพื่อที่เราจะไม่ต้องเสียเวลามานั่งออกแบบฟอนต์ขึ้นใหม่ ผู้เขียนจึงเลือกนำฟอนต์ของ CU Writer จากไฟล์ NORMAL.FON มาใช้

ลักษณะฟอนต์ของ CU Writer

ฟอนต์ของ CU Writer แต่ละตัวมีขนาด 20 ไบต์ ฟอนต์ไฟล์จึงมีขนาด



รูปที่ 1 แสดงการจัดลำดับฟอนต์ภาษาไทยของ CU Writer



รูปที่ 2 แสดงตารางบิตแม็พที่ออกแบบขึ้น

256×20 = 5120 ไบต์ แต่ละไบต์ใน 20 ไบต์จะเก็บบิตแม็พของแต่ละสแกนไลน์เอาไว้ โดยเราสามารถแบ่งฟอนต์ภาษาไทยของ CU Writer ออกได้เป็น 4 พวก คือ

1. วรรณยุกต์และการันต์ ได้แก่ ็ ่ ้ ๊ ๋ ์ ใช้สแกนไลน์ที่ 0-3 รวม 4 เส้น

2. สระบน ได้แก่ ิ ี ึ ื ั ใช้สแกนไลน์ที่ 3-6 รวม 4 เส้น

3. พยัญชนะและสระกลาง ได้แก่ ก-ฮ ฯ ะ า ำ เ แ โ ใ ไ ๅ ๆ ๐-๙ ฿ ๕ ใช้สแกนไลน์ที่ 5-18 รวม 14 เส้น

4. สระล่าง ได้แก่ ุ ู ฺ ใช้สแกนไลน์ที่ 16-19 รวม 4 เส้น

5. อักขระอื่น ๆ ใช้สแกนไลน์ที่ 0-19 ทุกเส้น ดังแสดงในรูปที่ 1

เนื่องจากฟอนต์ของ CU Writer เก็บอยู่ในฟอร์แมตเดียวกันกับของตารางบิตแม็พของจออีจีเอและวีจีเอ อีกทั้งขนาดของบิตแม็พของอักขระภาษาไทยมีขนาดใหญ่ที่สุด 14 ไบต์ ซึ่งเพียงพอที่บิตแม็พขนาด 14 ไบต์ของจออีจีเอ และขนาด 16 ไบต์ของจอวีจีเอ จะสามารถรองรับได้ เราจึงสามารถนำฟอนต์ของ CU Writer มาใช้งานได้โดยสะดวก

ออกแบบตารางบิตแม็พ

จากการที่การแสดงผลภาษาไทยจะต้องทำใน 3 ระดับ เราจึงต้องสร้างฟอนต์ที่เป็นการผสมระหว่างสระบนกับ

วรรณยุกต์เข้าด้วยกัน เท่าที่เป็นไปได้ ทั้งหมดผู้เขียนได้ออกแบบตารางบิตแม็พสำหรับอักขระกราฟิก 128 ตัวหลังในตารางแอสกี ดังแสดงในรูปที่ 2

สำหรับช่องที่ว่างผู้เขียนยังได้ปล่อยว่างเอาไว้ ผู้อ่านสามารถเพิ่มเติมได้เอง จะสังเกตเห็นได้ว่า ผู้เขียนได้เตรียมตัวอักขระสำหรับตีกรอบเอาไว้ให้ด้วย เกี่ยวกับค่ารหัสแอสกีของตัวตีกรอบนี้ ผู้เขียนได้ค้นพบความลับบางอย่างซ่อนอยู่

ความลับของตารางบิตแม็พ

จากการสังเกตของผู้เขียนพบว่า จำนวนบิตที่ใช้แสดงอักขระแต่ละตัวบนจอภาพเท่ากับ 9 บิต แต่จำนวนบิตที่ใช้เก็บฟอนต์ในบิตแม็พมีเพียง 8 บิต แล้วทำไมเราจึงสามารถใช้ตัวตีกรอบตีเส้นตามแนวนอนได้อย่างต่อเนื่อง โดยเส้นไม่ขาดช่วง ทั้งนี้เนื่องจากในการแสดงผลอักขระจำพวกตัวตีตารางและบล็อกแบบต่าง ๆ บิตที่ 8 จะถูกทำสำเนาไปใช้เป็นบิตที่ 9 เมื่อนำฟอนต์ออกแสดงทางจอภาพ ซึ่งอักขระชุดดังกล่าวได้แก่ อักขระหมายเลข 176-178 และอักขระหมายเลข 179-223 รวมทั้งหมด 48 ตัว แต่เนื่องจากอักขระภาษาไทยตามมรหัสสมอ. ได้จับจองรหัสแอสกีดังกล่าวเอาไว้เกือบหมดแล้ว คงเหลืออยู่แต่เพียงรหัสแอสกีหมายเลข 219-223 เท่านั้น ผู้เขียนจึงได้นำตัวตีกรอบซึ่งจำเป็นจะต้องมีความต่อเนื่องในบิตที่ 9 คือ ┌ └ ─ รวม 3 ตัว ไปเก็บอยู่ในตำแหน่งดังกล่าว

นำฟอนต์ CU Writer มาใช้งาน

เพื่อให้สะดวกในการนำไปใช้งาน และเพื่อให้โปรแกรมไม่ต้องทำการโหลดฟอนต์จากไฟล์ NORMAL.FON ทุกครั้งที่เริ่มรันโปรแกรม ผู้เขียนจึงได้แปลงรูปของฟอนต์ไฟล์ให้กลายเป็นอินคลูดไฟล์ ซึ่งทำการประกาศค่าคงที่เป็นตารางบิตแม็พของอักขระกราฟิก 128 ตัวหลังของตารางแอสกีแทน ดังในรูปที่ 3 โดยได้เก็บฟอนต์แต่ละพวกอยู่ในช่วงสแกนไลน์ต่าง ๆ กันดังนี้

รูปที่ 3 เป็นตารางบิตแม็พของอักขระกราฟิก 128 ตัวหลังของตารางแอสกี

```
/* VGAFON.I */

typedef unsigned char  font_t[16];
const font_t  font[128]  /* 128..255 */ ={
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 128 */
      0x02, 0x02, 0x02, 0x00, 0x00, 0x21, 0x51, 0x3E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 129 */
      0x31, 0x12, 0x3C, 0x00, 0x00, 0x21, 0x51, 0x3E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 130 */
      0x6D, 0x95, 0x52, 0x00, 0x00, 0x21, 0x51, 0x3E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 131 */
      0x04, 0x1F, 0x04, 0x00, 0x00, 0x21, 0x51, 0x3E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 132 */
      0x02, 0x02, 0x02, 0x00, 0x00, 0x3C, 0x42, 0x7E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 133 */
      0x31, 0x12, 0x3C, 0x00, 0x00, 0x3C, 0x42, 0x7E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 134 */
      0x6D, 0x95, 0x52, 0x00, 0x00, 0x3C, 0x42, 0x7E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 135 */
      0x04, 0x1F, 0x04, 0x00, 0x00, 0x3C, 0x42, 0x7E },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 136 */
      0x02, 0x02, 0x02, 0x00, 0x00, 0x79, 0x85, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 137 */
      0x31, 0x12, 0x3C, 0x00, 0x00, 0x79, 0x85, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 138 */
      0x6D, 0x95, 0x52, 0x00, 0x00, 0x79, 0x85, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 139 */
      0x04, 0x1F, 0x04, 0x00, 0x00, 0x79, 0x85, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 140 */
      0x02, 0x02, 0x02, 0x00, 0x00, 0x76, 0x8D, 0xFE },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 141 */
      0x31, 0x12, 0x3C, 0x00, 0x00, 0x76, 0x8D, 0xFE },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 142 */
      0x6D, 0x95, 0x52, 0x00, 0x00, 0x76, 0x8D, 0xFE },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 143 */
      0x04, 0x1F, 0x04, 0x00, 0x00, 0x76, 0x8D, 0xFE },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 144 */
      0x02, 0x02, 0x02, 0x00, 0x00, 0x75, 0x8D, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 145 */
      0x31, 0x12, 0x3C, 0x00, 0x00, 0x75, 0x8D, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 146 */
      0x6D, 0x95, 0x52, 0x00, 0x00, 0x75, 0x8D, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 147 */
      0x04, 0x1F, 0x04, 0x00, 0x00, 0x75, 0x8D, 0xFF },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 148 */
      0x03, 0x0C, 0x14, 0x08, 0x00, 0x3C, 0x42, 0x7E },
    { 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C,   /* 149 */
      0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 150 */
      0xFC, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C },
    { 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C,   /* 151 */
      0xFC, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 152 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 153 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 154 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 155 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 156 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 157 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 158 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 159 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 160 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x22, 0x42,   /* 161 */
      0x42, 0x42, 0x42, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x24, 0x54, 0x34, 0x14,   /* 162 */
      0x14, 0x14, 0x14, 0x18, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0xB4, 0xD4, 0x14, 0x14,   /* 163 */
      0x14, 0x14, 0x14, 0x18, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0x82, 0x92,   /* 164 */
      0xAA, 0xD2, 0x82, 0x82, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x6C, 0x92, 0x82, 0x92,   /* 165 */
      0xAA, 0xD2, 0x82, 0x82, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0xB2, 0xD2, 0x32, 0x22,   /* 166 */
      0x72, 0xAA, 0xA6, 0x62, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x08, 0x14, 0x0C, 0x04,   /* 167 */
      0x44. 0x24. 0x14. 0x0C, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0x12, 0x2A,   /* 168 */
      0x1A, 0x0A, 0x0A, 0x06, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x78, 0x84, 0x04, 0x44,   /* 169 */
      0xA4, 0x6E, 0x55, 0x66, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x02, 0x04, 0x28, 0x54, 0x34, 0x14,   /* 170 */
      0x14, 0x14, 0x14, 0x18, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x02, 0x04, 0xA8, 0xD4, 0x14, 0x24,   /* 171 */
      0x24, 0x24, 0x28, 0x30, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x72, 0x8A, 0x4A, 0x8A,   /* 172 */
      0x8A, 0x9A, 0xAE, 0xDA, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x72, 0x8A, 0x4A, 0x8A,   /* 173 */
      0x8A, 0xCA, 0xAA, 0xEE, 0x00, 0x08, 0x0E, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0xE2, 0x22,   /* 174 */
      0x22, 0x62, 0xA2, 0x43, 0x16, 0x2A, 0x36, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0x62, 0x22,   /* 175 */
      0x22, 0x62, 0xA2, 0x42, 0x02, 0xAA, 0x56, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3E, 0x40, 0x7C, 0x02,   /* 176 */
      0x1A, 0x1A, 0x06, 0x06, 0x00, 0x2A, 0x56, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0xB6, 0xDA, 0x32, 0x42,   /* 177 */
      0x42, 0x42, 0x42, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x52, 0xAA, 0x8A, 0x8A,   /* 178 */
      0xAA, 0xDE, 0xAA, 0x9A, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x64, 0x94, 0x54, 0x94,   /* 179 */
      0x94, 0x96, 0x9D, 0xD6, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x42, 0x52,   /* 180 */
      0x6A, 0x5A, 0x4A, 0x72, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x6C, 0x92, 0x92, 0xA2,   /* 181 */
      0xD2, 0xB2, 0x92, 0xE2, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x22, 0x42,   /* 182 */
      0x42, 0x62, 0x52, 0x22, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0xC4, 0xAA, 0x52, 0x62,   /* 183 */
      0x42, 0x42, 0x42, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3E, 0x50, 0x7C, 0x12,   /* 184 */
      0x12, 0x12, 0x12, 0x1E, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x64, 0xA4, 0x64, 0x24,   /* 185 */
      0x44, 0x4E, 0x55, 0x66, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x42, 0xA2, 0x62, 0x22,   /* 186 */
      0x22, 0x22, 0x22, 0x7E, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x02, 0x02, 0x02, 0x42, 0xA2, 0x62, 0x22,   /* 187 */
      0x22, 0x22, 0x22, 0x3E, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x22, 0x52, 0x62, 0x42,   /* 188 */
      0x42, 0x42, 0x5A, 0x66, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x02, 0x02, 0x02, 0x22, 0x52, 0x62, 0x42,   /* 189 */
      0x42, 0x42, 0x5A, 0x66, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x42, 0xA2, 0x62, 0x22,   /* 190 */
      0x2A, 0x2A, 0x36, 0x22, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x02, 0x02, 0x02, 0x42, 0xA2, 0x62, 0x22,   /* 191 */
      0x2A, 0x2A, 0x36, 0x22, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0xC2, 0x22,   /* 192 */
      0x22, 0x62, 0xA2, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x42, 0xA2, 0x62, 0x22,   /* 193 */
      0x22, 0x7A, 0xA6, 0x62, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x22, 0x52, 0x62, 0x42,   /* 194 */
      0x32, 0x42, 0x42, 0x7E, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3E, 0x60, 0x7C, 0x02,   /* 195 */
      0x02, 0x06, 0x0A, 0x04, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x22, 0x42,   /* 196 */
      0x42, 0x62, 0x52, 0x22, 0x02, 0x02, 0x02, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0x02, 0x3A,   /* 197 */
      0x46, 0x62, 0x52, 0x22, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0x82, 0xE2, 0x22,   /* 198 */
      0x22, 0x62, 0xA2, 0x42, 0x02, 0x02, 0x02, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x02, 0x02,   /* 199 */
      0x02, 0x06, 0x0A, 0x04, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x04, 0x08, 0x3C, 0x42, 0x42, 0x5A,   /* 200 */
      0x5A, 0x52, 0x52, 0x32, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x42, 0xA2, 0x62, 0x2A,   /* 201 */
      0x2F, 0x22, 0x22, 0x7E, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x02, 0x04, 0x7C, 0x82, 0x02, 0x3A,   /* 202 */
      0x46, 0x62, 0x52, 0x22, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x46, 0xAA, 0x64, 0x4E,   /* 203 */
      0x52, 0x62, 0x42, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x06, 0x0A, 0x07, 0x42, 0xA2, 0x62, 0x22,   /* 204 */
      0x42, 0x5A, 0x66, 0x42, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x7C, 0xC2, 0x02, 0x22,   /* 205 */
      0x52, 0x62, 0x42, 0x7C, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x02, 0x04, 0x7C, 0xCA, 0x72, 0x22,   /* 206 */
      0x52, 0x62, 0x42, 0x3C, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x64, 0x4C, 0x34,   /* 207 */
      0x04, 0x04, 0x04, 0x08, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x62, 0x3C,   /* 208 */
      0x00, 0x62, 0x3C, 0x00, 0x00, 0x00, 0x00, 0x00 },
    { 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 209 */
      0x00, 0x00, 0x00, 0x00, 0x00, 0x21, 0x51, 0x3E },
```

รูปที่ 3 (ต่อ)

```
{ 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x02, 0x02,   /* 210 */
  0x02, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00 },
{ 0x60, 0x90, 0x60, 0x00, 0x1C, 0x22, 0x02, 0x02,   /* 211 */
  0x02, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 212 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x7E },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 213 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x79, 0x85, 0xFF },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 214 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x76, 0x8D, 0xFE },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 215 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x75, 0x8D, 0xFF },
{ 0x00, 0x0C, 0x04, 0x04, 0x00, 0x00, 0x00, 0x00,   /* 216 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x1A, 0x0A, 0x0E, 0x00, 0x00, 0x00, 0x00,   /* 217 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x08, 0x14, 0x08, 0x00, 0x00, 0x00, 0x00,   /* 218 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 219 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 220 */
  0x0F, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C },
{ 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C, 0x0C,   /* 221 */
  0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 222 */
  0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 223 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x10, 0x10, 0x10, 0x10,   /* 224 */
  0x10, 0x18, 0x14, 0x08, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x48, 0x48, 0x48, 0x48,   /* 225 */
  0x48, 0x48, 0x6C, 0x6C, 0x00, 0x00, 0x00, 0x00 },
{ 0x3E, 0x40, 0xF8, 0x04, 0x04, 0x04, 0x04, 0x04,   /* 226 */
  0x04, 0x06, 0x05, 0x02, 0x00, 0x00, 0x00, 0x00 },
{ 0x78, 0x84, 0x94, 0x74, 0x04, 0x04, 0x04, 0x04,   /* 227 */
  0x04, 0x06, 0x05, 0x02, 0x00, 0x00, 0x00, 0x00 },
{ 0x8C, 0x54, 0x24, 0x04, 0x04, 0x04, 0x04, 0x04,   /* 228 */
  0x04, 0x06, 0x05, 0x02, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x02, 0x02,   /* 229 */
  0x02, 0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0xEC, 0x92, 0x02, 0x02,   /* 230 */
  0x02, 0x02, 0x02, 0x02, 0x02, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 231 */
  0x01, 0x3E, 0x48, 0x36, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 232 */
  0x02, 0x02, 0x02, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 233 */
  0x31, 0x12, 0x3C, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 234 */
  0x6D, 0x95, 0x52, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 235 */
  0x04, 0x1F, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 236 */
  0x03, 0x0C, 0x14, 0x08, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 237 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x0C, 0x12, 0x0C },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 238 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 239 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x42,   /* 240 */
  0x42, 0x42, 0x42, 0x3C, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x3C, 0x42, 0x4A,   /* 241 */
  0x32, 0x02, 0x04, 0x38, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x80, 0x94, 0xAA, 0xAA,   /* 242 */
  0xA2, 0xB2, 0x82, 0xFC, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x6C, 0xAA, 0x92,   /* 243 */
  0x92, 0xC2, 0xA2, 0x44, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x02, 0x3C, 0x40, 0x80,   /* 244 */
  0x98, 0x90, 0x88, 0x7C, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x02, 0x34, 0x78, 0x80,   /* 245 */
  0x98, 0x90, 0x88, 0x7C, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x80, 0x40, 0x3C, 0x02, 0x01,   /* 246 */
  0x21, 0x51, 0x62, 0x3C, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x02, 0x52, 0xAA, 0xAA,   /* 247 */
  0x8A, 0xCA, 0xAC, 0x48, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x02, 0x04, 0x78, 0x80,   /* 248 */
  0x86, 0x96, 0x92, 0x6C, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x01, 0x02, 0x1C, 0x74, 0x92,   /* 249 */
  0x88, 0xA4, 0xD2, 0x60, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0xCA, 0x9A, 0x6A,   /* 250 */
  0x0A, 0x0A, 0x0A, 0x14, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x40, 0x90,   /* 251 */
  0x92, 0x6D, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 252 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 253 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 254 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
{ 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,   /* 255 */
  0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 },
};
```

■

1. วรรณยุกต์และการันต์ อยู่ในสแกนไลน์เส้นที่ 8-11

2. สระบน อยู่ในสแกนไลน์เส้นที่ 12-15

3. พยัญชนะและสระกลาง อยู่ในสแกนไลน์เส้นที่ 1-14

4. สระล่าง อยู่ในสแกนไลน์เส้นที่ 0-3

หากต้องการนำตารางบิตแม็พดังกล่าวนี้ไปใช้กับจออีจีเอ ซึ่งใช้จำนวนสแกนไลน์ 14 เส้น ก็สามารถทำได้ดังนี้

1, 2. สำหรับวรรณยุกต์และสระบน รวมทั้งฟอนต์ที่เป็นการผสมวรรณยุกต์และสระบนเข้าด้วยกัน ให้ตัดไบต์บนสุดของบิตแม็พแต่ละตัวออก 3 ไบต์ แล้วแทรก 0 เป็นไบต์สุดท้าย (13)

3. พยัญชนะและสระกลาง ให้ตัดไบต์บนสุด (0) และไบต์ล่างสุด (15) ออก

4. สระล่าง ให้ตัดไบต์ล่างสุดของบิตแม็พแต่ละตัวออก 3 ไบต์ แล้วแทรก 0 เป็นไบต์บนสุด (0)

การนำฟอนต์ภาษาไทยออกแสดงผล

ขั้นตอนในการนำฟอนต์ภาษาไทยออกแสดงผลมีอยู่ 2 ขั้น ขั้นแรกเป็นการโหลดตารางบิตแม็พของเราเข้าไปเก็บในแรม แทนที่บิตแม็พของอักขระกราฟิก 128 ตัวหลังที่เราใช้อยู่เดิมก่อน หลังจากนั้นเราก็จะสามารถสั่งทำการแสดงผลออกทางจอภาพตามปกติได้เลย ไม่ว่าจะเป็นการเรียกใช้ฟังก์ชันของดอสไบออส หรือแม้แต่การยิงจอ (การเขียนลงวิดีโอแรมโดยตรง) ก็สามารถทำได้ทั้งสิ้น

สำหรับวิธีการโหลดตารางบิตแม็พของเราเข้าไปใช้งาน ทำได้โดยการเรียกใช้ไบออสอินเตอร์รัพต์ ฟังก์ชัน 11h บริการย่อยที่ 0 โดยผ่านค่าดังนี้

INT 10h-BIOS interrupt
Function 11h
Subservice 0-Load user-defined font (text modes)

AH = 11h

AL = 0

ES:BP = ชี้ไปยังตำแหน่งเริ่มต้นของตารางบิตแม็พของเรา

CX = จำนวนอักขระในตารางบิตแม็พ ในที่นี้เท่ากับ 128

DX = ออฟเซตในตารางบิตแม็พของจอภาพที่เราจะโหลดบิตแม็พของเราลงไป ในที่นี้เท่ากับ 128×16 = 2048 (หรือ 128×14 = 1792 สำหรับจออีจีเอ)

BL = หมายเลขตารางบิตแม็พที่ต้องการเปลี่ยน ในที่นี้เท่ากับ 0

BH = จำนวนไบต์ต่อบิตแม็พของอักขระ 1 ตัว ในที่นี้เท่ากับ 16 (หรือ 14 สำหรับจออีจีเอ)

ทั้งนี้ผู้เขียนได้ทำการเขียนเป็นฟังก์ชันเอาไว้ให้ผู้อ่านเรียกใช้ได้โดยง่าย ดังแสดงในรูปที่ 4 ผู้อ่านสามารถนำฟังก์ชันดังกล่าวนี้ไปโหลดฟอนต์ที่ผู้อ่านออกแบบขึ้นเองได้ตามใจชอบ จะกี่ตัวก็ได้ แอพพลิเคชันหลาย ๆ ตัวในเท็กซ์โหมดเช่น นอร์ตันยูทิลิตี้ ก็ใช้เทคนิคอันนี้ในการทำยูสเซอร์อินเตอร์เฟสที่สวยงามมากขึ้นเช่นกัน

ผลที่ได้คือภาษาไทย 8-25 บรรทัด

แม้ว่าเราจะต้องเป็นผู้ที่ทำหน้าที่จัดระดับภาษาไทยและทำการผสมสระ

```
/*=================================================================
  write_char_gen()
    changes character generator for 'n' characters starting
    from the character 'ch' to 'map'
=================================================================*/
void  write_char_gen( int ch, int n, char far *map )
{
  _CX = n;                    /* 'n' characters              */
  _DX = (unsigned char)ch;     /* start with character 'ch'   */
  _BX = 0x1000;                /* 16 bytes per character
                              using map number 0            */
  _ES = FP_SEG( map );         /* segment of new char gen     */
  _BP = FP_OFF( map );         /* offset of new char gen      */
  _AX = 0x1100;                /* func. 11h - load char gen
                                          sub func. 0 - user defined  */
  geninterrupt( 0x10 );        /* BIOS call                   */
}
```

รูปที่ 4 เป็นฟังก์ชันตัวอย่างเพื่อนำไปโหลดฟอนต์ที่ได้ออกแบบขึ้น

เอง แต่ก็คงไม่ยากจนเกินไปใช่ไหมครับ ทางที่ดีเราควรจะถ่ายสำเนาตารางในรูปที่ 2 เอาไว้ใช้ประกอบตอนเขียนโปรแกรมของเรา โดยอาจจะต้องโหลดไดรเวอร์ภาษาไทยเข้าช่วย แต่ผลพลอยได้ที่เราได้รับก็คือ เราอาจแสดงผลได้สูงสุดถึง 25 บรรทัด ถ้าไม่ใช้สระบน สระล่าง หรือวรรณยุกต์เลย หรือหากไม่มีสระล่างและสระบน (หรือวรรณยุกต์) ถูกใช้ในตำแหน่งเดียวกัน เช่น การแสดงข้อความ 2 บรรทัดต่อไปนี้

```
กญแจ  <--- บรรทัดที่ 1
 ,  ั  <--- บรรทัดที่ 2
ของฉน  <--- บรรทัดที่ 3
```

จะเห็นว่า ใช้เพียง 3 บรรทัดในการแสดงภาษาไทย 2 บรรทัด

```
/*=================================================
  box()
    draws a box
=================================================*/
void  box( int x1, int y1, int x2, int  y2 )
{
#define  S_VERT       149
#define  S_TOP_LEFT   220
#define  S_TOP_RIGHT  150
#define  S_BOT_LEFT   221
#define  S_BOT_RIGHT  151
#define  S_HORIZ      222

  int  x, y;
  struct text_info  t;

  gotoxy( x1, y1 );                  /* top          */
  putch( S_TOP_LEFT );
  for ( x=x1+1; x<x2; x++ )
    putch( S_HORIZ );
    putch( S_TOP_RIGHT );

    for ( y=y1+1; y<y2; y++ ) {          /* left & right */
      gotoxy( x1, y );
      putch( S_VERT );
      gotoxy( x2, y );
      putch( S_VERT );
    }

    gotoxy( x1, y2 );                  /* bottom       */
    putch( S_BOT_LEFT );
    for ( x=x1+1; x<x2; x++ )
      putch( S_HORIZ );

    /* outputs directly to Video RAM to prevent scrolling */
    gettextinfo( &t );
    poke( ( ( peekb( 0x0040, 0x0049 ) != 7 ) ?
              0xB800 : 0xB000 ), (y2-1)*160+((x2-1)<<1),
          S_BOT_RIGHT | ( t.attribute << 8 ) );
}
```

รูปที่ 5 เป็นฟังก์ชันสำหรับตีกรอบ

```
/*=================================================================
  restore_char_gen()
    restores old character generator from ROM
=================================================================*/
void  restore_char_gen()
{
  _BL = 0;                  /* using map number 0          */
  _AX = 0x0002;              /* func. 0 - set mode
                                        AL = text mode 2          */
  geninterrupt( 0x10 );      /* BIOS call                   */
}
```

รูปที่ 6 ฟังก์ชันตัวอย่างเพื่อให้ผู้อ่านเรียกใช้ได้โดยง่าย

สำหรับกรณีที่แย่ที่สุด จะแสดงภาษาไทยได้ 8 บรรทัด คือแสดงภาษาไทย 1 บรรทัดใช้จำนวนบรรทัดจริง ๆ 3 บรรทัด

ฟังก์ชันสำหรับตีกรอบ

เนื่องจากตัวอักขระตีกรอบ ไม่ได้อยู่ในตำแหน่งเดียวกันกับชุดฟอนต์ปกติที่เราใช้อยู่ ผู้เขียนจึงได้เขียนฟังก์ชันสำหรับตีกรอบขึ้นมาให้ใช้งานด้วย อย่ามองว่าเป็นรูทีนสำหรับการตีกรอบธรรมดา ๆ นะครับ เพราะฟังก์ชันนี้สามารถตีกรอบได้เต็มจอ โดยจอภาพไม่เลื่อนขึ้นด้วย กุญแจสำคัญอยู่ที่การเขียนมุมขวาล่างของกรอบ ซึ่งผู้เขียนใช้วิธีเขียนลงวิดีโอแรมโดยตรง โดยใช้สีปัจจุบันตามที่โปรแกรมเมอร์กำหนดไว้ล่าสุด ซึ่งพิจารณาได้จาก text_info ของไลบรารี conio ดังแสดงในรูปที่ 5

การคืนตารางบิตแม็พเดิม

ก่อนที่เราจะจบการทำงานโปรแกรมของเรา ก็อย่าลืมคืนตารางมิตแบ็พเดิมให้กับจอภาพนะครับ ซึ่งวิธีการนั้นก็ทำได้ง่าย ๆ โดยการเรียกใช้ไบออสอินเตอร์รัพต์ฟังก์ชัน 11h บริการย่อยที่ 4 เพื่อทำการโหลดตารางบิตแม็พจากรอมเข้ามาเก็บในตารางที่ต้องการ โดยผ่านค่าดังนี้

INT 10h-BIOS interrupt
Function 11h
Subservice 0-Load 8 × 16 ROM font (text modes)

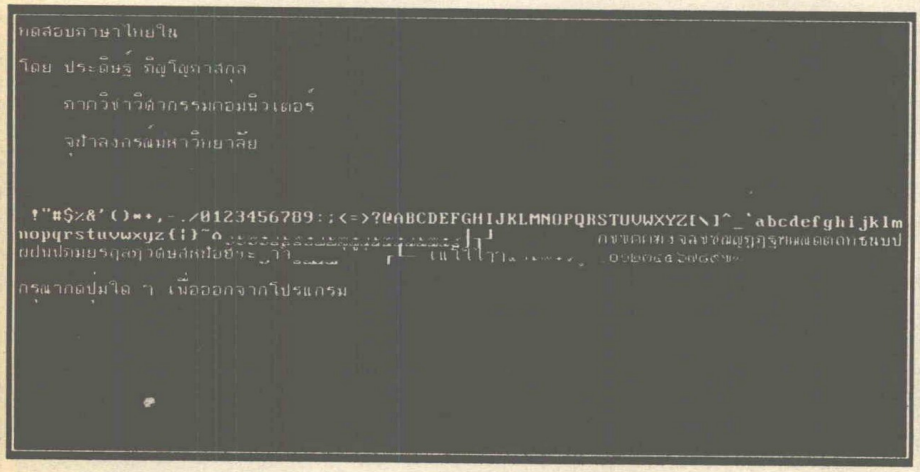
รูปที่ 7 แสดงโปรแกรมตัวอย่างที่นำไปใช้งาน

```
#include <conio.h>
#include <dos.h>
#include "VGAFON.I"

/*=================================================================
  main()
=================================================================*/
void  main()
{
  int  i;

  clrscr();

  /* loads Thai character generator for 128 graphic characters */
  write_char_gen( 128, 128, (char far *)font );

  /* displays Thai & English characters at the same time in text mode */
  textbackground( BLUE );
  textcolor( LIGHTGRAY );
  box( 1, 1, 80, 25 );                  /* draws full-screen box */
  window( 2, 2, 79, 24 );
  clrscr();

  textcolor( YELLOW );
  cputs( "ทดสอบภาษาไทยใน " );
  textcolor( LIGHTMAGENTA | BLINK );
  cputs( "text mode\n\r" );
  textcolor( LIGHTCYAN );
```

```
    cputs( "      \n\r" );          /* shows my name          */
    cputs( "โดย ประดษฐ ภญโญภาสกล\n\r" );
    cputs( "      \n\r" );
    cputs( "    ภาควชาวศวกรรมคอมพวเตอร\n\r" );
    cputs( "      \n\r" );
    cputs( "    จฬาลงกรณมหาวทยาลย\n\r" );
    cputs( "    \n\r" );

    textcolor( LIGHTRED );              /* shows date             */
    cputs( "๑ มกราคม ๒๕๓๖\n\r\n\r" );

    textcolor( WHITE );                 /* shows all characters   */
    for ( i=32; i<=255; i++ )
      putch( i );

    textcolor( LIGHTGREEN );            /* "press any key..."     */
    cputs( "\n\r          \n\r" );
    cputs( "กรณากดปมใด ๆ เพอออกจากโปรแกรม\n\r" );
    cputs( "    \n\r" );
    while ( kbhit() ) getch();          /* gets a key stroke      */
    while ( !getch() );
    textattr( LIGHTGRAY );
    clrscr();

    /* restores old character generator from ROM */
    restore_char_gen();
}
```

■

รูปที่ 7 (ต่อ)



รูปที่ 8 แสดงผลลัพธ์ที่ได้จากการรันโปรแกรมตัวอย่าง

AH = 11h

AL = 4 (หรือ 1 สำหรับจออีจีเอ ซึ่งจะเป็นการโหลดฟอนต์ขนาด 8×14 แทน)

BL = หมายเลขตารางบิตแม็พที่ต้องการเปลี่ยน ในที่นี้เท่ากับ 0

ทั้งนี้ผู้เขียนได้ทำการเขียนเป็นฟังก์ชันเอาไว้ให้ผู้อ่านเรียกใช้ได้โดยง่าย ดังแสดงในรูปที่ 6

ตัวอย่างการนำไปใช้งาน

ผู้เขียนได้ให้ตัวอย่างการนำไปใช้งานไว้ด้วย ดังแสดงในรูปที่ 7 ซึ่งผลที่ได้จากการรันโปรแกรมตัวอย่างจะเป็นดังในรูปที่ 8

บทสรุป

เห็นไหมครับว่าไม่ยากเลย เพียงโปรแกรมสั้น ๆ เราก็สามารถแสดงผลภาษาไทยในเท็กซ์โหมดได้แล้ว ผู้เขียนได้พยายามสอดแทรกเกร็ดความรู้และเทคนิคเล็ก ๆ น้อย ๆ เอาไว้ในโปรแกรมตัวอย่าง ซึ่งหวังว่าสิ่งเหล่านั้นผู้อ่านจะสามารถนำไปใช้ประโยชน์ได้ต่อไป นอกเหนือไปจากการแสดงผลภาษาไทยในเท็กซ์โหมดด้วย สำหรับผู้อ่านที่ไม่ต้องการคีย์โปรแกรมเอง ก็สามารถใช้บริการซอฟต์เมล์ได้ครับ